

IOWA STATE UNIVERSITY

Digital Repository

Computer Science Technical Reports

Computer Science

3-17-1994

A Faster Algorithm for the Perfect Phylogeny Problem when the Number of Characters is Fixed

Richa Agarwala
Iowa State University

David Fernández-Baca
Iowa State University, fernande@iastate.edu

Follow this and additional works at: http://lib.dr.iastate.edu/cs_techreports

 Part of the [Theory and Algorithms Commons](#)

Recommended Citation

Agarwala, Richa and Fernández-Baca, David, "A Faster Algorithm for the Perfect Phylogeny Problem when the Number of Characters is Fixed" (1994). *Computer Science Technical Reports*. 15.
http://lib.dr.iastate.edu/cs_techreports/15

This Article is brought to you for free and open access by the Computer Science at Iowa State University Digital Repository. It has been accepted for inclusion in Computer Science Technical Reports by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

A Faster Algorithm for the Perfect Phylogeny Problem when the Number of Characters is Fixed

Abstract

We present an algorithm for determining whether a set of species, described by the characters they exhibit, has a perfect phylogeny, assuming the maximum number of characters is fixed. This algorithm is simpler and faster than the known algorithms when the number of characters is at least 4.

Disciplines

Theory and Algorithms

A Faster Algorithm for the Perfect Phylogeny Problem when the number of Characters is Fixed

TR94-05

Richa Agarwala and David Fernandez-Baca

March 22, 1994

Iowa State University of Science and Technology
Department of Computer Science
226 Atanasoff
Ames, IA 50011

A Faster Algorithm for the Perfect Phylogeny Problem when the Number of Characters is Fixed

RICHA AGARWALA AND DAVID FERNÁNDEZ-BACA

Department of Computer Science, Iowa State University, Ames, IA 50011

March 17, 1994

Abstract

We present an algorithm for determining whether a set of species, described by the characters they exhibit, has a perfect phylogeny, assuming the maximum number of characters is fixed. This algorithm is simpler and faster than the known algorithms when the number of characters is at least 4.

1 Introduction

A fundamental problem in biology is that of inferring the evolutionary history of a set of species, each of which is specified by the set of *traits* or *characters* that it exhibits [6, 7, 10, 11]. Information about evolutionary history can be conveniently represented by an evolutionary or *phylogenetic* tree, often referred to simply as a *phylogeny*. In one of the standard models, the problem can be expressed mathematically as follows. Let $\mathcal{C} = \{1, \dots, m\}$ be the *character set*, and for every $c \in \mathcal{C}$, let $\mathcal{A}_c = \{1, \dots, r_c\}$ be the set of allowable *states* for character c . We write r to denote $\max_{c \in \mathcal{C}} r_c$. A *species* $\mathbf{s} = (s_1, \dots, s_m)$ is a vector such that $\mathbf{s} \in \mathcal{A}_1 \times \dots \times \mathcal{A}_m$; s_c is referred to as the *state of character c for \mathbf{s}* , or the *state of \mathbf{s} on character c* . We assume that if $i \in \mathcal{A}_c$, then there exists a species $\mathbf{s} \in \mathcal{S}$ such that $s_c = i$. A *phylogeny* for \mathcal{S} is a tree whose vertices are species and every leaf is in \mathcal{S} . A phylogeny T for \mathcal{S} is *perfect* if for each $c \in \mathcal{C}$ and each state r_c of that character, the nodes having state r_c on character c form a subtree of T . The *perfect phylogeny problem* is to determine whether a given set of species \mathcal{S} has a perfect phylogeny. If \mathcal{S} admits a perfect phylogeny, the set of characters \mathcal{C} is said to be *compatible*. We should point out that in the biology literature the perfect phylogeny problem is more commonly known as the *character compatibility problem* [5]. In this context, one is frequently interested in computing a maximal set of compatible characters, since, in practice, character sets tend to be incompatible.

The perfect phylogeny problem was shown to be NP-complete by Bodlaender et al. [3] and, independently, by Steel [13]. Linear time algorithms have been found for $m = 3$ [4, 8]. The perfect phylogeny problem is known to be polynomially equivalent to the problem of triangulating colored graphs [14] which led to a perfect phylogeny algorithm running in $O((rm)^{m+1} + nm^2)$ time [12]. This is polynomial for every fixed m but not very fast in practical terms. Bodlaender, Fellows, and Hallett [2] have shown that the

perfect phylogeny problem is hard for $W[2]$, implying that it is unlikely to be solved by an algorithm whose running time is of the form $O(f(m)rn)$ where f is an exponential function. In this paper, we present a $O((r - n/m)^m (rm)^2)$ algorithm, which is polynomial for every fixed m and is faster and simpler than the known algorithms for $m \geq 4$. Using the polynomial equivalence of triangulating colored graphs and perfect phylogeny [14], we get an algorithm with a running time of $O(m^{k+2}k^2)$ for triangulating a k -colored graph having m edges.

2 Preliminaries

We now state some definitions and preliminary results from [1].

Lemma 1 *A set of species \mathcal{S} has a perfect phylogeny if and only if every subset of \mathcal{S} has one.*

Definition 1 Suppose T is a perfect phylogeny for \mathcal{S} and let \mathbf{p} be some vertex in T . The state of \mathbf{p} on character c is *forced* if \mathbf{p} lies on the path between vertices \mathbf{a} and \mathbf{b} in \mathcal{S} such that $a_c = b_c$. (Observe that if this is the case, we must have $p_c = a_c = b_c$.)

Definition 2 Suppose $G \subset \mathcal{S}$ and let $G' = \mathcal{S} - G$. $\mathcal{D}(G)$, the set of *distinguishing characters* of G , is the set of all $c \in \mathcal{C}$ such that for every $\mathbf{a} \in G$ and every $\mathbf{b} \in G'$, $a_c \neq b_c$. $\mathcal{M}(G)$, the set of *common characters*, is $\mathcal{C} - \mathcal{D}(G)$.

Obviously, $\mathcal{D}(G) = \mathcal{D}(G')$ and $\mathcal{M}(G) = \mathcal{M}(G')$.

Definition 3 A pair (G, G') where $G \subset \mathcal{S}$ and $G' = \mathcal{S} - G$ is called a *split* if, for every character, the number of common character states between G and G' is at most one. A split (G, G') is a *c-split* if $\mathcal{D}(G) \neq \emptyset$. If (G, G') is a split (c-split), G and G' are called *clusters* (*c-clusters*).

Observe that we can determine whether a partition (G, G') of \mathcal{S} is a split in $O(nm)$ time. Note also that if G is a cluster but not a c-cluster, then $\mathcal{M}(G) = \mathcal{C}$.

Definition 4 Let (G, G') be a split. We say that (G, G') is of *type I* if there exists an $\mathbf{s} \in G$ such that for all $c \in \mathcal{M}(G)$, s_c equals the unique common state between G and G' on character c and $|G - \{\mathbf{s}\}|, |G'| \geq 1$. If (G, G') is of type I, we refer to \mathbf{s} as a *connecting species*. If (G, G') is not of type I, we say that it is of *type II*.

Lemma 2 *If all c-splits are of type II, then, in every perfect phylogeny T of \mathcal{S} , every species $s \in \mathcal{S}$ is a leaf in T .*

Definition 5 A cluster G is said to be *compatible* with a vector \mathbf{s} if for every $c \in \mathcal{M}(G)$, s_c equals the unique common state for character c between G and $\mathcal{S} - G$.

Definition 6 A *subphylogeny* T_G for a cluster G is a perfect phylogeny for $G \cup \{\mathbf{x}\}$, where \mathbf{x} is a node such that G is compatible with \mathbf{x} . Node \mathbf{x} is referred to as *connection* of T_G .

The next result implies that, in searching for a perfect phylogeny for \mathcal{S} , we can restrict our attention to perfect phylogenies constructed entirely from subphylogenies.

Lemma 3 *Suppose \mathcal{S} has no type I c-splits. Let G be a cluster. Then, G has a subphylogeny if and only if there exist pairwise disjoint c-clusters G_1, \dots, G_k and a vector \mathbf{x} such that (i) for every $c \in \mathcal{M}(G)$, x_c equals the (unique) common state for character c between G and $\mathcal{S} - G$, (ii) $\cup_{i=1}^k G_i = G$, and (iii) each G_i is compatible with \mathbf{x} and has a subphylogeny.*

Corollary 4 *Suppose \mathcal{S} has no type I c-splits. \mathcal{S} has a perfect phylogeny if and only if there exist pairwise disjoint c-clusters G_1, \dots, G_k and a vector \mathbf{x} such that (i) $\cup_{i=1}^k G_i = \mathcal{S}$ and (ii) each G_i is compatible with \mathbf{x} and has a subphylogeny.*

We assume that there are no duplicate species and for every character, there are at least two character states which are exhibited by at least two species. The latter assumption can be made without loss of generality. To prove this, suppose that there exists a character $c \in \mathcal{C}$ such that at most one of its states p_c is exhibited by more than two species in \mathcal{S} . Assume without loss of generality that $c = m$. We claim that a perfect phylogeny T for \mathcal{S} on \mathcal{C} exists if and only if a perfect phylogeny T' exists for \mathcal{S} on $\mathcal{C} - \{c\}$. Deriving T' from T is easy. T can be derived from T' as follows.

$$\begin{aligned} V(T) &= \{(u_1, \dots, u_{m-1}, p_m) : (u_1, \dots, u_{m-1}) \in V(T')\} \cup \{\mathbf{s} : \mathbf{s} \in \mathcal{S}, s_m \neq p_m\} \\ E(T) &= \{(\mathbf{u}, \mathbf{v}) : ((u_1, \dots, u_{m-1}), (v_1, \dots, v_{m-1})) \in E(T')\} \\ &\quad \cup \{(\mathbf{s}, (s_1, \dots, s_{m-1}, p_m)) : \mathbf{s} \in \mathcal{S}, s_m \neq p_m\}. \end{aligned}$$

The next section describes a way of finding the c-clusters which are compatible with a vector \mathbf{x} and can give a perfect phylogeny for \mathcal{S} by using Corollary 4. Section 4 gives the basic algorithm for finding a perfect phylogeny for \mathcal{S} . The algorithm builds subphylogenies for the c-clusters found using the method presented in Section 3 and efficiently searches for pairwise disjoint c-clusters G_1, \dots, G_k and a vector \mathbf{x} satisfying the conditions of Corollary 4. Section 5 improves on the basic algorithm by considering fewer c-clusters.

3 Finding c-clusters

Given a species \mathbf{x} , Kannan and Warnow [9] defined an equivalence relation $E_{\mathbf{x}}$ as the transitive closure of the following relation \mathcal{R} on $\mathcal{S} - \{\mathbf{x}\}$:

$$(\mathbf{a}, \mathbf{b}) \in \mathcal{R} \text{ if there exists } c \in \mathcal{C} \text{ such that } a_c = b_c \neq x_c.$$

It is clear from this definition that two species in \mathcal{S} which are in the same equivalence class must be in the same component of $T - \{\mathbf{x}\}$, for any perfect phylogeny T of $\mathcal{S} \cup \{\mathbf{x}\}$. The set of equivalence classes is denoted by $(\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}$ and can be computed in $O(nm)$ time [9]. Note that if \mathbf{x} is an internal node in any perfect phylogeny on \mathcal{S} , then $|(\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}| \geq 2$. In particular, we make the following remark.

Proposition 5 *If $|(\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}| = 1$ for every $\mathbf{x} \in \mathcal{S}$, then \mathcal{S} has no type I splits.*

We now reformulate a result in [9].

Lemma 6 *Let $G \in (\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}$. If $\mathcal{S} \cup \{\mathbf{x}\}$ has a perfect phylogeny, then G is a c-cluster and it is compatible with \mathbf{x} .*

Proof: From the definition of $E\mathbf{x}$, it follows that G is a cluster and G is compatible with \mathbf{x} . We now show that each $G \in (\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}$ is a c-cluster. The claim is trivially true if $G = \mathcal{S}$. Otherwise, it suffices to show that there exists a perfect phylogeny with no duplicate nodes for $\mathcal{S} \cup \{\mathbf{x}\}$ such that the species in each component of $T - \{\mathbf{x}\}$ give us the equivalence classes of $(\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}$.

By Lemma 1 there exists a perfect phylogeny for each $G \cup \{\mathbf{x}\}$ where $G \subset \mathcal{S}$. Specifically, let T_1, \dots, T_k be perfect phylogenies with no duplicate nodes for $G_1 \cup \{\mathbf{x}\}, \dots, G_k \cup \{\mathbf{x}\}$, where $(\mathcal{S} - \{\mathbf{x}\})/\mathbf{x} = \{G_1, \dots, G_k\}$. Since each G_i is compatible with \mathbf{x} , identifying the nodes for \mathbf{x} from each T_i gives us a perfect phylogeny for $\mathcal{S} \cup \{\mathbf{x}\}$ with no duplicate nodes. To prove the claim, we need to show that \mathbf{x} is a leaf in each T_i . Suppose this is not true. Let H_1, \dots, H_l be the species in components of $T_i - \{\mathbf{x}\}$. As G_i is an equivalence class of $(\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}$, there exists a species $\mathbf{p} \in H_j, \mathbf{q} \in H_{j'}$ for some $j \neq j'$, and a character $c \in \mathcal{C}$ such that $p_c = q_c \neq x_c$. But this implies that T_i is not a perfect phylogeny, which gives us a contradiction. \square

Corollary 7 *Let $G \in (\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}$. If $\mathcal{S} \cup \{\mathbf{x}\}$ has a perfect phylogeny, then $G \cup \{\mathbf{x}\}$ has a perfect phylogeny and \mathbf{x} is a leaf in every perfect phylogeny of $G \cup \{\mathbf{x}\}$.*

Lemma 8 *Suppose \mathcal{S} has no type-I c-splits and $\mathcal{S} \cup \{\mathbf{x}\}$ has a perfect phylogeny for some $\mathbf{x} \notin \mathcal{S}$. Then \mathcal{S}/\mathbf{x} is a set of c-clusters satisfying the conditions of Corollary 4.*

Proof: The result is a consequence of the fact that $\mathbf{x} \notin \mathcal{S}$ and the following:

1. Each $G \in \mathcal{S}/\mathbf{x}$ is a c-cluster and is compatible with \mathbf{x} . This follows from Lemma 6.
2. The set of c-clusters \mathcal{S}/\mathbf{x} are pairwise disjoint and their union equals \mathcal{S} . This is because $E\mathbf{x}$ is an equivalence relation on \mathcal{S} .
3. Each $G \in \mathcal{S}/\mathbf{x}$ has a subphylogeny. To prove this, note that since \mathbf{x} is a connection for G , it is sufficient to show that $G \cup \{\mathbf{x}\}$ has a perfect phylogeny. This follows from Lemma 1, since $G \cup \{\mathbf{x}\} \subseteq \mathcal{S} \cup \{\mathbf{x}\}$ and $\mathcal{S} \cup \{\mathbf{x}\}$ has a perfect phylogeny. \square

The c-clusters of interest are the ones which can give us a perfect phylogeny for \mathcal{S} using Corollary 4 and Lemma 8. As there are $\prod_{i=1}^m r_i$ choices for \mathbf{x} and each \mathcal{S}/\mathbf{x} can have at most n classes, the total number of c-clusters of interest is $O(r^m n)$; all such clusters can be found in $O(r^m nm)$ time.

4 The Basic Algorithm

The maximum number of edges in any perfect phylogeny with no duplicate nodes is at most $(r - 1)m$ because in a perfect phylogeny, nodes having the same character state on any character form a subtree. This gives us the following necessary but insufficient condition for deciding existence of a perfect phylogeny for \mathcal{S} .

Proposition 9 *If $n > (r - 1)m + 1$, then \mathcal{S} has no perfect phylogeny.*

We now present an algorithm which constructs a perfect phylogeny, if it exists, such that the adjacent nodes differ in exactly one character state. The steps carried out by the basic algorithm are as follows.

Step 0. If $n > (r - 1)m + 1$, then **Return FAILURE**.

Step 1. Find if there exists a species $\mathbf{x} \in \mathcal{S}$ such that $|(\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}| \geq 2$. If there is any such species, we get subproblems $G \cup \{\mathbf{x}\}$ for each $G \in (\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}$. By Lemma 1, \mathcal{S} has a perfect phylogeny if and only if each of these subproblems has one. The details of this construction are given in [1].

We can now assume that for each species $\mathbf{x} \in \mathcal{S}$, $|(\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}| = 1$. From Proposition 5, \mathcal{S} has no type I c-splits. Hence, a species \mathbf{x} is a candidate for an internal node in any perfect phylogeny for \mathcal{S} only if $\mathbf{x} \notin \mathcal{S}$ and $|(\mathcal{S} - \{\mathbf{x}\})/\mathbf{x}| \geq 2$.

Step 2. For each species $\mathbf{x} \notin \mathcal{S}$, find \mathcal{S}/\mathbf{x} . Create a directed search graph W such that

$$\begin{aligned} V(W) &= \{[\mathcal{S}, \mathbf{x}] : \mathbf{x} \notin \mathcal{S}, |\mathcal{S}/\mathbf{x}| \geq 2\} \cup \{[G, \mathbf{x}] : \mathbf{x} \notin \mathcal{S}, |\mathcal{S}/\mathbf{x}| \geq 2, G \in \mathcal{S}/\mathbf{x}\} \\ E(W) &= \{([G, \mathbf{x}], [\mathcal{S}, \mathbf{x}]) : [G, \mathbf{x}] \in V(W)\} \cup \{([G_1, \mathbf{x}_1], [G_2, \mathbf{x}_2]) : G_1 \subseteq G_2 \\ &\quad \text{and } \mathbf{x}_1 \text{ differs from } \mathbf{x}_2 \text{ in exactly one character state}\} \end{aligned}$$

Each node of W will have an associated boolean variable, initially FALSE.

Step 3. Assign the value TRUE to every $[G, \mathbf{x}] \in V(W)$ such that $|G| = 1$.

A node $w = [G, \mathbf{x}]$ with $|G| \geq 2$ is said to be *active* if every $w' \in V(W)$ such that $(w', w) \in E(W)$ has a truth value assigned to it.

Do the following until every active node has a truth value.

1. Choose any active node w that has no truth value assigned to it.
2. Make w TRUE if there exists a vector \mathbf{x} such that $[G', \mathbf{x}]$ is TRUE for every $[G', \mathbf{x}]$ such that $([G', \mathbf{x}], w) \in E(W)$. Otherwise, make w FALSE.

Step 4. If there exists a node $[\mathcal{S}, \mathbf{x}]$ which is TRUE, **return SUCCESS**. Otherwise, **return FAILURE**.

All the steps can be done in $O(r^{m+1}nm)$ time as the maximum number of edges entering a node is $O(rnm)$ and the maximum number of outgoing edges is $O(rm)$.

Lemma 10 *If there exists a perfect phylogeny for $\mathcal{S} \cup \{\mathbf{x}\}$ then $[G, \mathbf{x}]$ is assigned TRUE for each $G \in \mathcal{S}/\mathbf{x}$.*

Proof: The proof is by induction on $|G|$. The base case for nodes $[G, \mathbf{x}]$ where $|G| = 1$ is true from Step 3 and the fact that G is compatible with \mathbf{x} . Suppose the claim holds for all nodes $[G, \mathbf{x}]$ where $|G| < k$. Consider a node $[G, \mathbf{x}]$ where $|G| = k$. Let T be a perfect phylogeny for $G \cup \{\mathbf{x}\}$. By Corollary 7, \mathbf{x} is a leaf in T . Let \mathbf{y} be the first node on the path starting at \mathbf{x} in T with degree at least 3. There exists at least one such node as there are no type I splits and $|G| \geq 2$. It is now easy to see that there exist pairwise disjoint clusters in \mathcal{S}/\mathbf{y} whose union gives G , each of which has a subphylogeny. By induction hypothesis, the corresponding nodes of W will be assigned TRUE. Thus, $[G, \mathbf{y}]$ will be TRUE and this value will be propagated to every node $[G, \mathbf{p}]$ such that \mathbf{p} is on the path between \mathbf{y} and \mathbf{x} in T . Hence, $[G, \mathbf{x}]$ will eventually become TRUE. \square

Theorem 11 *\mathcal{S} has a perfect phylogeny if and only if there exists a node $[\mathcal{S}, \mathbf{x}]$ for some \mathbf{x} which is TRUE.*

Proof: Follows from Lemma 8, Lemma 10, and Steps 2 and 3 of the basic algorithm. \square

Constructing the perfect phylogeny: If the algorithm returns SUCCESS, we can build a perfect phylogeny by traversing W and constructing an in-tree D as follows. Choose any node $[\mathcal{S}, \mathbf{a}]$ having value TRUE as the root of D . Next, do the following until every leaf $[G, \mathbf{x}]$ in D has $|G| = 1$.

Pick any leaf $w = [G, \mathbf{x}]$ in D such that $|G| \geq 2$. Let A_w be any set of node such that
(i) A_w contains all $w' = [G', \mathbf{y}] \in V(W)$ such that $(w', w) \in E(W)$, for some $\mathbf{y} \notin \mathcal{S}$ and
(ii) for every $w' \in A_w$, w' has been assigned the value TRUE. (Intuitively, A_w is a set of nodes that led to w being assigned the value TRUE.) Add to D all the nodes in A_w , as well as the edges (w', w) such that $w' \in A_w$.

After D has been constructed, we can build a perfect phylogeny T by disregarding the orientation of the edges of D and replacing each node $[G, \mathbf{x}]$ of D with a node labeled \mathbf{x} .

5 An Improved Algorithm

The following lemma gives a way of reducing the number of internal nodes that need to be considered for building a perfect phylogeny.

Lemma 12 *Suppose there exists a species $\mathbf{s} \in \mathcal{S}$ and a character $c \in \mathcal{C}$ such that for every other $\mathbf{s}' \in \mathcal{S}$, $s_c \neq s'_c$; i.e., s_c is unique to species \mathbf{s} . If there exists a perfect phylogeny for \mathcal{S} , then there exists a perfect phylogeny for \mathcal{S} such that no node except \mathbf{s} has state s_c for character c .*

Proof: State s_c is never forced for any node $\mathbf{x} \neq \mathbf{s}$, so a different assignment is possible for x_c . Hence, as long as least one character state for character c is exhibited by more than one species, we can obtain a perfect phylogeny for \mathcal{S} such that no node except \mathbf{s} has state s_c for character c . \square

If \mathcal{S} has no type I splits, it is easy to see that for every species $\mathbf{s} \in \mathcal{S}$, there exists a character $c \in \mathcal{C}$ such that for every other $\mathbf{s}' \in \mathcal{S}$, $s_c \neq s'_c$. For each $\mathbf{s} \in \mathcal{S}$, pick one such character. Let k_c be the number of times character c is chosen in this process. From the above lemma, the number of internal nodes that need to be considered is $\prod_{i=1}^m (r - k_i)$ and in Step 2 of the algorithm in Section 4, we only need to consider the c -clusters derived from these internal nodes. As $\sum_{i=1}^m k_i = n$ and $\prod_{i=1}^m (r - k_i) \leq (r - \sum_{i=1}^m k_i/m)^m$, the running time of this improved algorithm is $O((r - n/m)^m(rnm))$. Since $n \leq (r - 1)m + 1$ by Proposition 9, the running time of the new algorithm is $O((r - n/m)^m(rm)^2)$.

Acknowledgements

We thank Tandy Warnow and Sampath Kannan for providing us with the equivalence relation $E_{\mathbf{x}}$ and their results in [9].

References

- [1] R. AGARWALA AND D. FERNÁNDEZ-BACA, *A Polynomial-Time Algorithm for the Perfect Phylogeny Problem when the Number of Character States is Fixed*, In Proceedings of the 34th Annual Symposium on Fundamentals of Computer Science (1993), pp. 140–147.
- [2] H. BODLAENDER, M. FELLOWS, AND M. T. HALLETT, *The parameterized complexity of DNA mapping, perfect phylogeny and other problems of “bounded width”*, manuscript.
- [3] H. BODLAENDER, M. FELLOWS, AND T. WARNOW, *Two strikes against perfect phylogeny*, In Proceedings of the 19th International Colloquium on Automata, Languages, and Programming, Springer Verlag, Lecture Notes in Computer Science (1992), pp. 273–283.
- [4] H. BODLAENDER AND T. KLOKS, *A simple linear time algorithm for triangulating three-colored graphs*, In Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science (1992), pp. 415–423.
- [5] W. H. E. DAY AND D. SANKOFF, *Computational complexity of inferring phylogenies by compatibility*, Syst. Zool., Vol. 35, No. 2 (1986), pp. 224–229.
- [6] G. F. ESTABROOK, *Cladistic methodology: A discussion of the theoretical basis for the induction of evolutionary history*, Annual Review of Ecology and Systematics, 3 (1972), pp. 427–456.
- [7] W. M. FITCH, *Aspects of molecular evolution*, Annual Reviews of Genetics, 7 (1973), pp. 343–380.
- [8] S. KANNAN AND T. WARNOW, *Triangulating three-colored graphs*, SIAM J. on Discrete Mathematics, 5 (1992), pp. 249–258.
- [9] S. KANNAN AND T. WARNOW, Personal Communication.
- [10] W. J. LE QUESNE, *A method of selection of characters in numerical taxonomy*, Syst. Zool., 18 (1969), pp. 201–205.
- [11] W. J. LE QUESNE, *Further studies based on the uniquely derived character concept*, Syst. Zool., 21 (1972), pp. 281–288.
- [12] F. R. MCMORRIS, T. WARNOW, AND T. WIMER, *Triangulating vertex colored graphs*, In Proceedings of the 4th Annual Symposium on Discrete Algorithms (1993), Austin, Texas.
- [13] M. A. STEEL, *The complexity of reconstructing trees from qualitative characters and subtrees*, Journal of Classification, 9 (1992), pp. 91–116.
- [14] T. J. WARNOW, *Combinatorial Algorithms for Constructing Phylogenetic Trees*, Ph.D. Thesis, University of California, Berkeley, May 1991.



IOWA STATE UNIVERSITY

OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

SCIENCE
with
PRACTICE

Tech Report: TR94-05
Submission Date: March 22, 1994